An Efficient Tag Identification Algorithm Based on Improved Collision Detection

Ya-Ning Yan[1] and Jian Xiong[2]

1. ZTE Telecommunications College, Xi'an Peihua University, Xi'an 710065, China

2. School of Control Science and Engineering, Dalian Univeristy of Technology, Dalian 116024, China

394811297@qq.com, sunli256@126.com.

***Abstract* — Tag collision has a negative impact on the performance of RFID systems. In this paper, we propose an algorithm termed anti-collision protocol based on improved collision detection (ACP-ICD). In this protocol, dual prefixes matching and collision bit detection technique are employed to reduce the number of queries and promptly identify tags. According to the dual prefixes matching method and collision bit detection in the process of collision arbitration, idle slots are eliminated. Moreover, the reader makes full use of collision to improve identification efficiency. Both analytical and simulation results are presented to show that the performance of ACP-ICD outperforms existing anti-collision algorithms.**

Keywords* — *RFID; anti-collision; Aloha; DPMM.

I. INTRODUCTION

Multiple tags collision is a vital challenge in radio frequency identification (RFID) systems, where multiple tags within single reader's operation range share a common wireless channel to communicate with the reader. If more than one tag transmit messages to the reader simultaneously, signals from the tags will collide and corrupt each other, causing the well-known multiple tags collision problem [1]. The tags collision problem substantially degrades the identification performance, especially in tags density RFID environment. To cope with this problem, many anti-collision algorithms [2-9] have been proposed in the literature. Technically, these algorithms can be classified into three groups: tree-based [2-4], Aloha-based [5-7] and hybrid protocols [8-9][11].

Tree-based algorithms are considered deterministic and provide simple tag designs, e.g., the Collision Tree (CT) [3] and Query Tree (QT) [2], repeatedly separate collided tags into disjoint subsets until a set has a single tag whose identification can be obtained without collisions. This procedure is repeated until all tags are identified successfully. These methods are also called memoryless protocols, because the current response of each tag depends on the current prefix of the reader but not on the past history of the reader's queries [3]. Both of CT and QT use tag IDs to achieve collision detection. As a result, it causes a large identification delay. Aloha-based algorithms, also called probabilistic or random access protocols because tags use random numbers to respond. These protocols suffer from significant performance degradation when the number of unread tags varies in a large scale, and cannot avoid the so called tag starvation problem [10].

Tree-based and Aloha-based protocols are combined in hybrid protocols to avoid their respective disadvantages. Most of them first implement a binary splitting procedure or an estimation procedure to obtain approximate tag backlog (unread tags), then combine a variation of Aloha or tree protocol to reduce the identification delay. Generally, these hybrid protocols [8-9][11] can provide relatively better performance than tree-based and Aloha-based protocols at the expense of a complex reader and tag designs.

In this paper, we propose a new simple but efficient hybrid tag anti-collision algorithm, termed anti-collision protocol based on improved collision detection (ACP-ICD). The algorithm introduces the idea of bit tracking technology and dual prefix matching into collision arbitration mechanism based on EPC C1 Gen2 RFID system. The brand new collision arbitration mechanism is that the reader repeatedly queries tags' SID (serial ID) so as to split tags into small subsets until there is at most two tags in each subset. It can provide deterministic identification and lower communication overhead via dual prefixes matching method. Theoretical analysis and simulation indicate that our proposed ACP-ICD outperforms existing anti-collision protocols.

II. RELATED WORDS

A Bit Tracking Technology

Bit tracking technology is commonly based on Manchester code [3-4][10-11], which defines the value of a bit as a voltage transition. A bit '0' is coded by a positive transition and a bit '1' by a negative transition. Since a 'no transition' state is not allowed in Manchester coding, the reader can identify and trace the collision to an individual bit. In this way, all bits of the received signal can be perceived accordingly. Similarly, in our proposed algorithm, we use bit tracking technology based on FM0 code since

EPC C1 Gen2 standard does not support Manchester code.

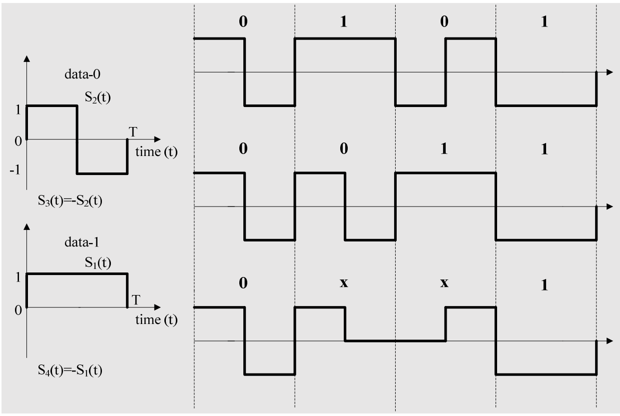

Figure1. Example of FM0 code

Fig. 1 shows an example of FM0 code. The IDs of tag 1 and tag 2 are "0101" and "0011", respectively. When tags 1 and 2 send their IDs simultaneously using the FM0 coding method, the interfered signal received by the reader is "0xx1", where "x" represents a collided bit. In this example, the locations of the collided bits are the second, and third bits. This collision information helps the reader separate the collided tags into subsets and identify the tags more quickly.

B. Dual Prefixes Matching Method

In traditional tree-based algorithms, the reader uses tag IDs to split tags into two subsets. The reader transmits a Probe command including a bit string called prefix. Every tag whose ID matches the prefix respond to the reader with its remaining ID. In these algorithms, a stack is used as a prefix pool to hold prefixes for next probes. Initially, an empty string ε is pushed into the stack and all tags respond to the reader. After that the reader continues to expand the prefixes until every tag responds individually. When the stack is exhausted, the reader can conclude that all the tags have been identified and terminates the identification process. In our proposed solution, we use dual prefixes matching method (DPMM) to eliminate the idle slots and save the transmitted bits.

The essential idea of DPMM is to assign two prefixes (only one bit difference) to each collided slot when the reader detects the collision occurs. In a slot, tags first match their serial IDs (SID is a sequence in 16bits length equaling to the RN16 of tags based EPC C1 Gen2) with the prefixes. When the first prefix matched, the tags response the rest part of the tag's full SID except the prefix part. And once the second prefix matched, the tags response the rest part of the tag's full SID except the prefix part after a r-bits time delay, where r equals to the length of full SID subtracts that of prefix. If collision happens, the reader updates the prefix stack and recursively identifies the colliding tags.

C. Collision Detection and Identification

In Aloha-based algorithms, the reader issues a Query command which contains a Q parameter to set the frame size as 2Q. Then each tag receiving the Query command randomly generates a value from 0 to 2Q-1 and load the value into its slot counter (SC). Only the tag whose SC is zero responds to the reader by sending an RN16, a 16-bit random number, within time T1. If the reader receives the RN16 without collision, it sends back an ACK command which contains the same RN16 within time T2. If the tag's RN16 is the same as that in the ACK command, it sends its EPC to the reader. If collision occurs, the reader sends a QueryRep command to start a next slot. The QueryRep command is used to enable all tags to decrement their SC by 1. The reader adjusts the frame size at the end of current frame until all tags are identified successfully. The collision detection method of Aloha-based algorithm is RN16-based generated by the tags. It only detects whether collision happens or not by utilizing 16bits data. In tree-based algorithms, the reader uses tags' IDs coded by Manchester code to detect and trace the collision, which is described in the section 2.1. The features of this collision detection method is efficient but requires large data transmitted in the process of detection. Combining the advantage of collision detection methods of two kinds of algorithms, we propose an enhanced collision detection method. We assign a unique SID for each tag during the manufacture. The length of SID is 16bits, which can support 216 tags. The SID is used to detect and trace the collision by utilizing bit tracking technology based on FM0 code. Noting that the SID can be stored in the tag's user memory.

D. Collision Detection and Identification

In this sub-section, we introduce the existing state-of-art works of anti-collision algorithms, which include collision tree (CT) [3], dynamic binary tree slotted Aloha (DBTSA) [9], Improved Linearized Combinatorial Model (ILCM) [7], and optimal binary tracking tree protocol (OBTT) [11].

CT is based on QT and belongs to the deterministic protocol. The main characteristic of CT is that both generating prefixes and splitting tags are according to the first collided bit. The CT can effectively eliminate the idle slots generated during the reader's query procedure and improve the performance of a binary query algorithm: the system efficiency is maintained at around 50%. ILCM is a low complexity Aloha-based algorithm, it only brings a modest floating point operations (FLOP) cost, and can be easily implemented as a tag backlog estimation method. It significantly reduces the computation energy consumption of the estimation, nonetheless, its performance is deteriorated with the increasing number of tags. DBTSA and OBTT are hybrid algorithms which enhance the identification performance by combining the advantages of tree-based and Aloha-based protocols at the expense of complex hardware architecture of the reader and tags. In DBTSA, tags are randomly assigned to slots of a frame and if some tags collide in a slot, the collided slots in the slot will be resolved by binary tree splitting while the other tags in the subsequent slots will wait. The system efficiency of DBTSA is about 40%. OBTT employs a bit estimation algorithm to partition tags into small groups, and then uses a query tracking tree to quickly identify tags. The system efficiency of OBTT is close to 61.4%, which is the highest so far. However, these hybrid algorithms have complicate implementation and high identification delay.

III. ALGORITHM DESCRIPTION

In this section, we introduce the proposed ACP-ICD in detail. Before describing the proposed algorithm, we first introduce the system transmission model between the reader and tags. Considering practical applications, we use the similar transmission model defined in the industrial standard, i.e., EPC C1 Gen2. Fig. 2 illustrates the link timing of data exchange between the reader and tags for traditional Aloha-based, tree-based algorithm and our proposed ACP-ICD. Where T1 is the time from reader transmission to tag response, T2 is the time from tag response to reader transmission, T3 is the guard time among multiple SIDs or EPCs. As can be observed in Fig. 2, although the time duration of successful slot in our scheme is longer than that of other algorithms, the whole identification efficiency can be improved. Functionally, ACP-ICD is based on QT and belongs to the deterministic protocol, but with essential difference. The main characteristic of ACP-ICD is that splitting collided tags are according to tag's SID and DPMM. In the ACP-ICD, the reader holds a stack to store prefixes for next query. If collision happens, the reader updates the stack according to the collided bits detection and uses a command to probe tags in the next slot. The identification process ends until the stack is empty. For prefix P1P2...Pk and response R1R2...Rc-1Rc, where Pi, Ri is a binary value 0 or 1, and Rc is the highest collide bit, the reader uses P1P2...PkR1R2...Rc-1Rc-110 as new prefix, where P1P2...PkR1R2...Rc-1Rc-1 is a common prefix, 1 is pre1 and 0 is pre2 (pre1-1). In the next slot, the tags matched with P1P2...PkR1R2...Rc-1Rc-11 first respond to the reader and then the tags matched with P1P2...PkR1R2...Rc-1Rc-10 will respond to the reader after a delay. Meanwhile, in ACP-ICD, the tags only transmit their SIDs except the part which is the same as the received prefix. Fig. 3 shows the flow diagram of ACP-ICD, where COM_STR denotes the common prefix.

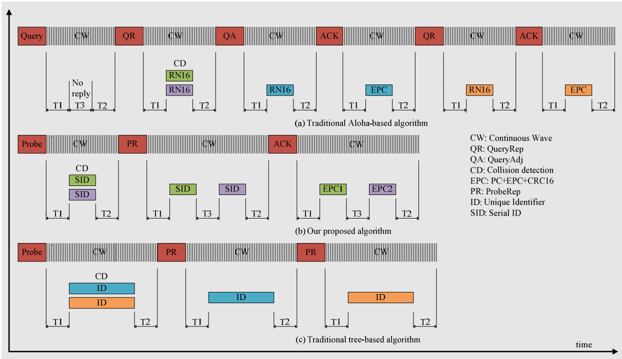

Figure2. Link timing between the reader and tags under various algorithms

According to the principle of the proposed algorithm, there are three kinds of slot status in our scheme.

Successful slot: when a single tag's response is received or two tags' responses matched with each prefix are received. In the traditional tree-based algorithms, a successful slot corresponds to an identified tag. In ACP-ICD, the number of tags can be identified is more than one.

Collided slot: when multiple tags respond to the same prefix with different IDs, which causes the reader cannot identify any tags.

Identifiable collision slot: when the reader receives multiple IDs in a same slot and obtain a single ID with a correct CRC checksum. Under this case, the reader can identify a tag.

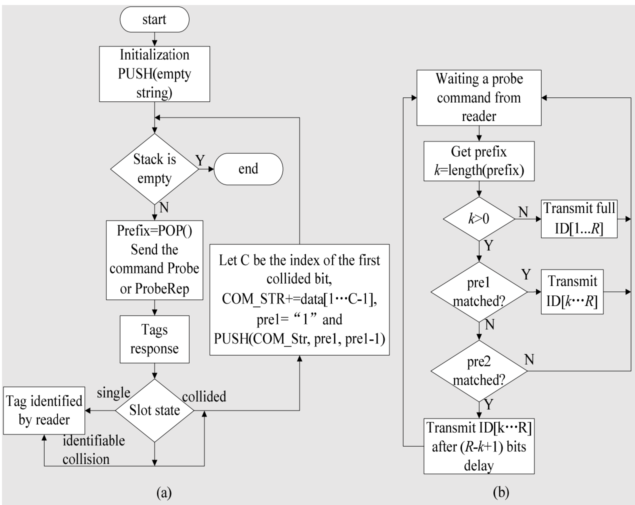

Figure.3. The flowchart of ACP-ICD: (a) for reader (b) for tags

It is noted that we consider dual-prefix due to the simplicity and the tradeoff between the performance and architecture complexity of tags, however, the proposed method can be extended to Multi-prefix cases.

Once detecting the collision, the reader updates the prefixes and stores them into the stack. The identification procedure will be repeated until the stack is empty. Considering there are n tags waiting to be identified in the operation range of a reader, we can conclude the following lemma.

Lemma 1: If the number of tags to be identified is n, then the number of total queries (E(Q)) is n.

Proof: According to the principle of ACP-ICD, the E(Q) can be expressed as:

$$E(Q) = 1 + C_1 + 2 * C_2 \quad (1)$$

where C1 and C2 represent the occurrence number of single bit collision and consecutive bits collision, respectively. In this paper, we only consider the single bit collision. Thus, C2 is equal to 0. According the analysis of CT [3], C1 is equal to n-1. Hence, the E(Q) in our proposed algorithm can be given as:

$$E(Q) = 1 + n - 1 = n \quad (2)$$

According to the definition of traditional system efficiency, the efficiency of ACP-ICD is n/n=100%. As can be observed from Fig. 2, the time duration of a slot is type-dependent. Clearly, the performance metric of system efficiency is unreasonable due to the discrepancy in different slot type's duration in various anti-collision algorithm. Consequently, we will introduce new performance evaluation metrics named identification rate, time efficiency in the next section.

IV. SIMULATION RESULTS

We evaluate the time efficiency, coordination time, identification rate, and communication overhead of the proposed ACP-ICD algorithm, and compare its performance with existing Aloha-based, tree-based and Hybrid-based methods, including CT [3], ILCM [7], DBTSA [9], and OBTT [11] over extensive Monte Carlo simulations. The time efficiency can be defined as

$$T_{effi} = \frac{T_{ness}}{T_{total}} \quad (3)$$

where Tness is the necessary time of valid data (EPC or ID) transmission, Ttotal is the total time to identify all tags. Note that the total time for identifying tags consist of Tness and coordination time such as the time duration of commands, guard time, etc. The identification rate is the number of tags identified by a reader per second. Compared to the traditional system efficiency metric, time efficiency and identification rate are more suitable to evaluate the performance of an anti-collision algorithm in terms of identification time.

TAB 1. PARAMETERS VALUES FOR NUMERICAL COMPUTATIONS

Parameters	value	parameter	value
Reader-to-tag data-0	1Tari	RTcal	37.5us
Reader-to-tag data-1	2Tari	TRcal	50us
Reader-to-tag rate	80kbps	T1	62.5us
Tag-to-reader rate	160kbps	T2	62.5us
Tpri	6.25us	T3	25us
Tari	12.5us	RN16	16bits
SID	16bits	EPC	128bits
ID	96bits	Query	22 bits
QueryAdj	9bits	QueryRep	4bits
Ack	18bits	Probe	16bits
ProbeRep	16bits	ID	128bits
EAck	18/34bits	DR	8

To obtain the time efficiency and identification rate, we need to calculate the time intervals of every step and command used in the identification process. The primary parameters used in the simulations are listed in Table I. The parameters are set according to the EPC C1 Gen2 standard.

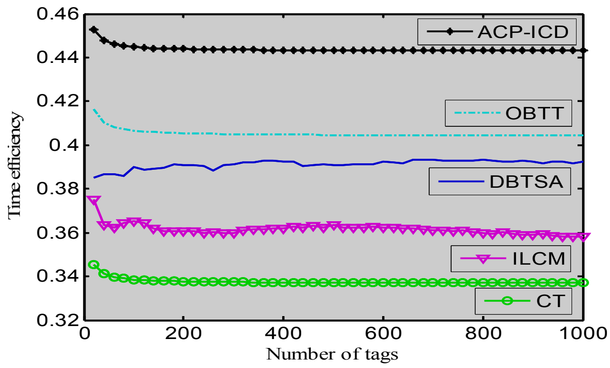

Fig.ure 4. Simulation results: time efficiency

Fig. 4 depicts the comparison of time efficiency between reference methods and ACP-ICD when the number of tags varies from 20 to 1000. The figure indicates that the time efficiency of ACP-ICD is always better than that of the existing approaches. Specifically, ACP-ICD achieves about 0.4438, whereas DBSTA, CT, ILCM, and OBTT achieve about 0.3912, 0.3376, 0.3615, and 0.4052, respectively. As can be observed from Fig.4, the time efficiency of all algorithms is below 0.5, which is far less than the traditional system efficiency can be achieved. The reason is that the coordination time during the identification process is very long and cannot be avoided.

Although the system efficiency of CT is higher than Aloha-based algorithm ILCM and Hybrid-based algorithm DBTSA, the time efficiency of CT is lowest. Since CT is a QT-based algorithm, it uses ID to perform collision arbitration and costs a large amount of time. By contrast, our proposed ACP-ICD decreases the coordination time by using DPMM and SID. It will not introduce extra time load because the length of SID is equal to that of RN16 in

Aloha-based algorithm. Fig. 5 shows the simulation results for the coordination time required to identify all tags. To identify 1000 tags, the proposed ACP-ICD spends about 1 second, whereas BTSA, CT, ILCM, and OBTT spend 1.24, 1.57, 1.43 and 1.18 seconds, respectively. Compared with other methods, the ACP-ICD consumes least coordination time by using bit tracking technology and DPMM. As a result, it improves the identification performance of algorithm. The average identification rate of various algorithms is plotted in Fig. 6.

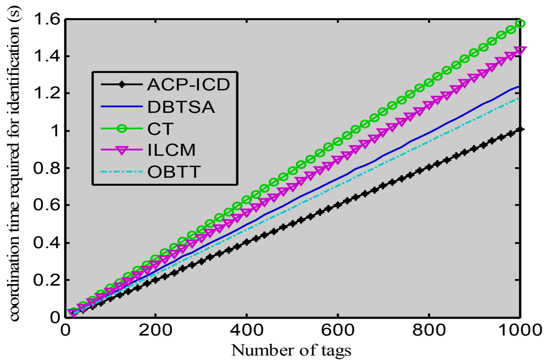

Figure.5. Simulation results: coordination time

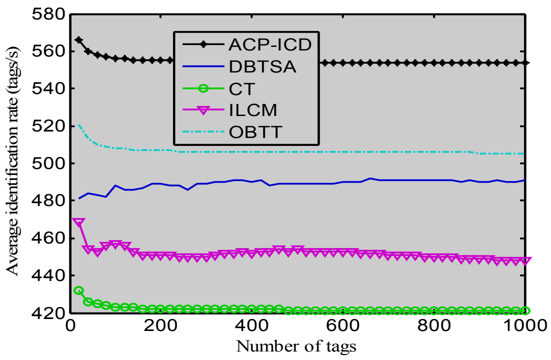

Figure 6. Simulation results: average identification rate.

V. CONCLUSION

In this paper, we proposed a novel deterministic anti-collision algorithm for identifying multiple RFID tags within the operation range of a reader. By using dual prefixes matching method and SID to perform collision detection and identification, the performance of identification is significantly improved. Through theoretical analysis and simulation results, we verified that ACP-ICD outperforms existing state-of-art algorithms in terms of time efficiency, coordination time saving, identification rate and communication overhead.

REFERENCE

[1] M. Bolic, D. Somplot-Ryl, I. Stojmenovic, "RFID systems: research trends and challenges," John Willey and Sons, 2010.

[2] J. Su, G. Wen, and J. Han, "An efficient RFID anti-collision algorithm for ISO 18000-6B protocol," Acta Electronic Sinica, vol. 42, no. 12, pp. 2515-2519, 2014.

[3] X. Jia, Q. Feng, and L. Yu, "Stability analysis of an efficient anti-collision protocol for RFID tag identification," IEEE Trans. Commu., vol. 60, no. 8, pp. 2285-2294, Aug. 2012.

[4] L. Zhang, J. Zhang, X. Tang, "Assigned tree slotted Aloha RFID Tag anti-collision protocols," IEEE Trans. Wireless Commun., vol. 12, no. 11, pp. 5493-5505, Nov. 2013.

[5] W.-T. Chen, "An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length ALOHA," IEEE Trans. Autom. Sci. Eng., vol. 6, no. 1, pp. 9-15, Jan. 2009.

[6] EPCglobal:'EPC radio-frequency identify protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz-960 MHz', version 2.0.0, 2013.

[7] Petar Solic, Joskoi Radic, and Nikola Rozic, "Energy efficient tag estimation method for ALOHA-based RFID systems," IEEE Sensors J., vol. 14, no. 10, pp. 3637-3647, Oct. 2014.

[8] T.F.L. Porta, G. Maselli, and C. Petrioli, "Anticollision protocols for single-reader RFID systems: temporal analysis and optimization," IEEE Trans. Mob. Comput., vol. 10, no. 2, pp. 267-279, Feb. 2011.

[9] H. Wu, Y. Zeng, J. Feng, and Y. Gu, "Binary tree slotted ALOHA for passive RFID tag anti-collision," IEEE Trans. Parall. Distrib. Sys., vol. 24, no. 1, pp. 19-31, Jan. 2013.

[10] Y. H. Chen, S. J. Horng, R. S. Run, J. L. Lai, R. J. Chen, W. C. Chen, Y. Pan, and T. Takao, "A novel anti-collision algorithm in RFID systems for identifying passive tags," IEEE Trans Ind. Informat., vol. 6, no. 1, pp. 105–121, Feb. 2010.

[11] Y. C. Lai, L. Y. Hsiao, and B. S. Lin, "Optimal slot assignment for binary tracking tree protocol in RFID tag identification,"IEEE/ACM Transactions on Networking, vol. 23, no. 1, p p. 255–268, Feb. 2015.